IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventors:	Mathew C. Mattina et al.	Examiner: Charles Anya
Serial No.:	09/924,934	Group Art Unit: 2194
Filed:	August 8, 2001	Docket No.: 200302133-1
Title:	Mechanism for Handling Load Lock/Store Conditional Primitives in Directory-Based Distributed Shared Memory Multiprocessors	

APPEAL BRIEF UNDER 37 C.F.R. § 41.37

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Appeal Brief is filed in response to the Final Office Action mailed April 4, 2008 and Notice of Appeal filed on August 4, 2008.

AUTHORIZATION TO DEBIT ACCOUNT

It is believed that no extensions of time or fees are required, beyond those that may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required (including fees for net addition of claims) are hereby authorized to be charged to Hewlett-Packard Development Company's deposit account no. 08-2025.

I. REAL PARTY IN INTEREST

The real party in interest is Hewlett-Packard Development Company, LP, a limited partnership established under the laws of the State of Texas and having a principal place of business at 20555 S.H. 249 Houston, TX 77070, U.S.A. (hereinafter "HPDC" or "Appellants"). HPDC is a Texas limited partnership and is a wholly-owned affiliate of Hewlett-Packard Company, a Delaware Corporation, headquartered in Palo Alto, CA. The general or managing partner of HPDC is HPQ Holdings, LLC. Assignments are recorded at Reel/Frame 012067/0828 and 014628/0103.

II. RELATED APPEALS AND INTERFERENCES

There are no known related appeals, judicial proceedings, or interferences known to Appellants, the Appellants' legal representative, or assignee that will directly affect or be directly affected by or have a bearing on the Appeal Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1 – 16, 18, 19, 21 – 26, and 28 – 34 are pending in the application. Claims 1 – 5, 7 – 9, 13 – 16, 18, 19, 21, 22, 24 – 26, and 28 – 31 are finally rejected. Claims 6, 10 – 12, 23, and 32 – 34 are objected to but would be allowed if written in independent form. Claim 17, 20, and 27 are withdrawn. The rejection of claims 1 – 5, 7 – 9, 13 – 16, 18, 19, 21, 22, 24 – 26, and 28 – 31 is appealed.

IV. STATUS OF AMENDMENTS

In the final office action mailed 04/04/2008, claims 1, 13, and 24 were objected to as ending in a comma. In an after-final response mailed 06/04/2008, claims 1, 13, and 24 were amended to end in a period. In an advisory action mailed 08/05/2008, the examiner indicated that the amendments were entered. All amendments have been entered.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The following provides a concise explanation of the subject matter defined in each of the claims involved in the appeal, referring to the specification by page and line number and to the drawings by reference characters, as required by 37 C.F.R. § 41.37(c)(1)(v). Each element of the claims is identified by a corresponding reference to the specification and drawings where applicable. Note that the citation to passages in the specification and drawings for each claim element does not imply that the limitations from the specification and drawings should be read into the corresponding claim element or that these are the sole sources in the specification supporting the claim features.

Paragraph [0017] in the Brief Summary of the Invention section describes a multiprocessor distributed shared memory system in which a processor obtains exclusive control of a data block by asserting a Load Lock signal to the Home processor that controls that data block If the processor with exclusive control does not complete operations on the data block prior to the time that the data block is displaced from the cache of that processor, it issues a Victim To Shared message which indicates to the Home processor that it should remain a sharer of the data block. In the event that another processor seeks exclusive rights to the same data block, the Home processor issues an Invalidate message to what was previously the Owner processor. This causes that processor to recognize that the Load Lock/Store Conditional operation pair has failed. If no Invalidate message is received, and when the processor executes the Store Conditional instruction, it seeks exclusive control of the data block by issuing a Read with Modify Intent Store Conditional message to the Home processor. If that processor is still a sharer, a writeable copy of the data block is sent to that processor, who completes modification of the data block. If the Home processor does not recognize that processor as a sharer of that data block, this indicates to the Home processor that an intervening write has occurred, and the Home processor returns a Store Conditional Failure message.

Claim 1 recites a distributed multiprocessing computer system (Fig. 2 shows a multiprocessor distributed processing system, #90), which includes a plurality of processors (Fig. 2, #100) each coupled to an associated memory module (Fig. 2, #125), wherein each of the associated memory modules stores data that is shared between said processors (Fig. 2, p. 12, ¶30, lines 1-9). The system comprises a Home processor (The

term "Home processor" means a processor that manages a memory and directory for a particular data block: see ¶26, lines 3-4 on p. 11) that includes a memory block and a directory for said memory block in the associated memory module (¶26, lines 3-4 on p. 11). An Owner processor (The term "Owner processor" means a processor that manages an exclusive copy of a data block in its cache: see ¶26, lines 4-5 on p. 11) that includes a cache memory (Fig. 2, #115, such as L1 and L2 cache: see ¶ 31, lines 4-6 on p. 13), and wherein said Owner processor obtains an exclusive copy of said memory block, and stores said exclusive copy of said memory block in said cache memory (¶26, lines 4-5 on p. 11); and wherein said Owner processor begins write operations on said memory block and then displaces the exclusive copy of said memory block from said cache memory prior to completing the write operations on said memory block, and in response to displacing said memory block prior to completing the write operations said Owner processor returns said displaced copy of said memory block to said Home processor with a signal indicating that said Owner processor remains a sharer of said memory block (see ¶17, lines 3-16 on p. 8).

Claim 13 recites a method of maintaining memory coherence in a distributed shared memory computer system including a plurality of processors (Fig. 2; ¶30, lines 1-9 on p. 12). The method comprises requesting a copy of a memory block from a Home processor to perform a write operation on the copy of the memory block (¶18, lines 1-3 on p. 9); storing said copy of said memory block exclusively in a cache memory associated with an Owner processor (¶18, lines 3-4 on p. 9); updating a coherence directory for said memory block in said Home processor to indicate the write operation on the copy of said memory block in said cache memory associated with the Owner processor (¶32 on pages 13-14); displacing said copy of said memory block from said cache memory prior to completion of said write operation on said memory block (¶17, lines 3-7 on p. 8); in response to commencing but not completing said write operation, transmitting a message from said Owner processor to said Home processor relinquishing exclusive control of said memory block, said message indicating that said Owner processor should still be deemed a sharer of said memory block (¶17, lines 3-7 on p. 8; ¶36, lines 3-12 on p. 16).

Claim 15 recites the method of claim 13, wherein the Load Lock instruction forms part of a Load Lock/Store Conditional instruction pair (¶17 lines 9-10 on p. 8; and ¶38 on p. 18).

Claim 21 recites a method of maintaining memory coherence in a distributed shared memory computer system including a plurality of processors (Fig. 2; ¶30, lines 1-9 on p. 12). The method comprises requesting a copy of a memory block from a Home processor to perform write operations on the copy of the memory block (¶18, lines 1-3 on p. 9); storing said copy of said memory block exclusively in a cache memory associated with an Owner processor (¶18, lines 3-4 on p. 9); updating a coherence directory for said memory block in said Home processor to indicate the write operation on the copy of said memory block in said cache memory associated with the Owner processor (¶32 on pages 13-14); displacing said copy of said memory block from said cache memory prior to completion of the write operations on said memory block (¶17, lines 3-7 on p. 8); in response to commencing but not completing said write operations, transmitting a message from said Owner processor to said Home processor relinquishing exclusive control of said memory block, said message indicating that said Owner processor should still be deemed a sharer of said memory block; asserting a request to again obtain an exclusive copy of said memory block, wherein, in response to the request to again obtain an exclusive copy of the memory block, the Home processor determines if the Owner processor is a sharer of the memory block, and if so, the Home processor sends an exclusive copy of the memory block to the Owner processor (¶17, lines 3-7 on p. 8; ¶36, lines 3-12 on p. 16).

Claim 24 recites a distributed multiprocessing computer system (Fig. 2 shows a multiprocessor distributed processing system, #90). The system comprises a first processor (Fig. 2, #100a: the Home processor) that includes a memory block (Fig. 2, #125a) and a directory associated with said memory block that tracks a status of said memory block; a second processor (Fig. 2, #100b:the Owner processor) that includes a cache memory (Fig. 2, #115b, and wherein said second processor requests an exclusive copy of said memory block and stores said memory block-in said cache memory (¶26, lines 4-5 on p. 11); and wherein said second processor begins processing of said memory block and then displaces the exclusive copy of said memory block prior to completing the

processing of said memory block, and in response to displacing said memory block but not completing the processing said second processor transmits a signal to said first processor indicating that said second processor relinquishes exclusive control of said memory block but should remain a sharer of said memory block (see ¶17, lines 3-16 on p. 8).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1, 13, 16 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 5,887,138 to Hagersten et al. in view of U.S. Pat. No. 6,275,907 to Baumgartner et al.

Claims 2, 3, 14, 15, 21, 22, 25 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 5,887,138 to Hagersten et al. in view of U.S. Pat. No. 6,275,907 to Baumgartner et al. as applied to claims 1, 13 or 24 above, and further in view of U.S. Pat. No. 6,141,734 B1 to Razdan et al.

Claims 4, 5, 7, 8, 18, 19, and 28-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 5,887,138 to Hagersten et al. in view of U.S. Pat. No. 6,275,907 to Baumgartner et al. and further in view of U.S. Pat. No. 6,141,734 B1 to Razdan et al. and further in view of U.S. Pat. No. 6,185,658 B1 to Arimilli et al.

Claims 9 and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 5,887,138 to Hagersten et al. in view of U.S. Pat. No. 6,275,907 to Baumgartner et al. and further in view of U.S. Pat. No. 6,141,734 B1 to Razdan et al. and further in view of U.S. Pat. No. 6,185,658 B1 to Arimilli et al and further in view of U.S. Pat. No. 6,266, 744 to Hughes et al.

VII. ARGUMENT

The rejection of claims 1 – 5, 7 – 9, 13 – 16, 18, 19, 21, 22, 24 – 26, and 28 – 31 is improper, and Appellants respectfully requests withdraw of this rejection.

The claims do not stand or fall together. Instead, Appellants present separate arguments for various independent and dependent claims. Each of these arguments is separately argued below and presented with separate headings and sub-heading as required by 37 C.F.R. § 41.37(c)(1)(vii).

Overview of Claims and Primary References (Hagersten and Baumgartner)

As a precursor to the arguments, Appellants provide an overview of the claims and the primary reference (Hagersten and Baumgartner). This overview will assist in determining the scope and content of the prior art as required in Graham (see *Graham v. John Deere Co. of Kansas City*, 383 U.S. 1, 17-18 setting out an objective analysis for applying 103 rejections).

In one embodiment of the invention, processor A gets exclusive write operations on a memory block of another processor B. The processor A begins the write operation, does not complete it, and relinquished exclusive control of the memory block back to processor B. Importantly, processor A then gets sharer status (instead of no status). For example as recited in claim 1, the Owner processor receives an exclusive copy of data from a memory block of the Home processor. The Owner processor then displaces this exclusive copy and returns it to the Home processor. When the exclusive copy of the data is returned to the Home processor, a signal instructs the Home processor that the Owner processor **remains a sharer of the memory block**. In other words, the Owner processor is able to displace data from memory of the Home processor while maintaining a role that it does not in fact have (i.e., the role of a processor that continues to hold a copy of data in memory).

Hagersten teaches a multiprocessing computer system with processing nodes that determine whether an address of a memory transaction is a global address or a local address. If the address is global, then a NUMA coherency request is initiated. If the address is local, then COMA coherency request is initiated.

Baumgartner teaches a plurality of processing nodes with a home processing node having a shared memory and a coherence directory. The coherency directory indicates possible coherence states of copies of memory granules among a plurality of memory granules that are stored within at least one processing node other than the home processing node. A coherence mechanism manages processor reservations even in cases in which a reserving processor's cache hierarchy does not hold a copy of the reserved memory granule.

Claim Rejections: 35 USC § 103(a)

Claims 1, 13, 16 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 5,887,138 to Hagersten et al. in view of U.S. Pat. No. 6,275,907 to Baumgartner et al. These rejections are traversed.

Claims 1, 13, 16 and 24 recite one or more elements that are not taught or suggested in the art. These missing elements show that the differences between the combined teachings in the art and the recitations in the claims are great. As such, the pending claims are not a predictable variation of the art to one of ordinary skill in the art. Some examples are provided below with respect to claim 1.

As one example, independent claim 1 recites that the Owner processor begins write operations on said memory block and then displaces the exclusive copy of said memory block from said cache memory prior to completing the write operations on said memory block. The claim then recites that **in response to displacing said memory block prior to completing the write operations**, the Owner processor returns the displaced copy of said memory block to said Home processor with a signal indicating that said Owner processor **remains a sharer of said memory block**. The art does not teach or suggest these elements.

Hagersten teaches a multiprocessing computer system with processing nodes that determine whether an address of a memory transaction is a global or local address. Hagersten also discusses different coherency states (modified, owned, shared, and invalid) employed by the computer system. Nowhere does Hagersten teach or suggest that a processing node displaces an exclusive copy of memory prior to completing write operations on the memory and then **in response to displacing the memory prior to**

completing such write operations returns a signal indicating that the processing node **remains a sharer of memory**.

In fact, the examiner admits the following:

> Hagersten is silent with reference to wherein said Owner processor beings write operations on said memory block and then displaces the exclusive copy of said memory block from said cache memory prior to completing the write operations said Owner processor returns said displaced copy of said memory block to said Home processor with a signal indication that said Owner processor remains a sharer of said memory block. (See Final OA mailed 04/04/2008at pages 3-4).

Appellants agree with this admission. The examiner, however, attempts to cure this deficiency with Baumgartner, citing column 2, lines 52-55 and column 16, lines 16-24. Appellants respectfully disagree.

Baumgartner teaches a home processing node and a remote processing node. When the remote processing node has a reservation for a cache line that is not resident in its cache hierarchy, a coherence directory at the home processing node associates the cache line with a coherence state that indicates that the reserved cache line may be possibly held non-exclusively at the remote processing node (see column 2, lines 41-47). The section cited by the examiner (i.e., column 2, lines 52-55) expounds on this concept and further explains that the coherence state of the reserved cache line is set to the shared state in response to a writeback transaction from the remote processing node to the home processing node. The examiner also cites column 16, lines 16-24 at Baumgartner. This section is directed to claim 3 and portions of claim 2 which claim the concept of a writeback transaction.

These sections of Baumgartner have nothing to do with displacing "exclusive copies of memory" as recited in claim 1. Furthermore, these sections of Baumgartner have nothing whatsoever to do with a processing node that displaces an exclusive copy of memory prior to completing write operations on the memory and then **in response to displacing the memory prior to completing such write operations** returns a signal

indicating that the processing node **remains a sharer of memory**. In Baumgartner, no such signal is transmitted. Instead, the writeback transaction in Baumgartner is used to set the coherence state of the cache line to non-exclusive or shared. The signal is not indicating that the remote processor remains a sharer of memory. Again, the signal prompts the coherence state to be set. Furthermore, Baumgartner never teaches that the action of transmitting the writeback is "in response to displacing memory prior to completing write operations" as recited in claim 1.

The other cited art of record has at least the deficiencies that they do not teach or suggest a processing node displaces an exclusive copy of memory prior to completing write operations on the memory and then in response to displacing the memory prior to completing such write operations returns a signal indicating that the processing node remains a sharer of memory.

The differences between the claims and the teachings in the art are great since the cited references fail to teach or suggest all of the claim elements. As such, the pending claims are not a predictable variation of the art to one of ordinary skill in the art.

For at least these reasons, claims 1, 13, 16 and 24 are allowable over the cited art of record.

Claim Rejections: 35 USC § 103(a)

Claims 2, 3, 14, 15, 21, 22, 25 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 5,887,138 to Hagersten et al. in view of U.S. Pat. No. 6,275,907 to Baumgartner et al. as applied to claims 1, 13 or 24 above, and further in view of U.S. Pat. No. 6,141,734 B1 to Razdan et al. These rejections are traversed.

Claims 2, 3, 14, 15, 21, 22, 25 and 26 recite one or more elements that are not taught or suggested in the art. These missing elements show that the differences between the combined teachings in the art and the recitations in the claims are great. As such, the pending claims are not a predictable variation of the art to one of ordinary skill in the art. Some examples are provided below with respect to claim 21.

As one example, independent claim 21 recites that the Owner processor begins write operations on the memory block and then (in response to commencing but not completing the write operations) transmits a message to the Home processing node

relinquishing exclusive control of the memory block. The message also indicates that the Owner processor should still be deemed a sharer of the memory block. The art does not teach or suggest these elements.

The examiner admits that Hagersten does not teach or suggest these claim recitations:

> Hagersten is silent with reference to wherein said Owner processor beings write operations on said memory block and then displaces the exclusive copy of said memory block from said cache memory prior to completing the write operations said Owner processor returns said displaced copy of said memory block to said Home processor with a signal indication that said Owner processor remains a sharer of said memory block. (See Final OA mailed 04/04/2008at pages 3-4).

Appellants agree with this admission. The examiner, however, attempts to cure this deficiency with Baumgartner, citing column 2, lines 52-55 and column 16, lines 16-24 (on page 6 at number 11 of the final OA the examiner indicates same rejection as applied to claims 1, 5, and 9). Appellants respectfully disagree.

Baumgartner teaches a home processing node and a remote processing node. When the remote processing node has a reservation for a cache line that is not resident in its cache hierarchy, a coherence directory at the home processing node associates the cache line with a coherence state that indicates that the reserved cache line may be possibly held non-exclusively at the remote processing node (see column 2, lines 41-47). The section cited by the examiner (i.e., column 2, lines 52-55) expounds on this concept and further explains that the coherence state of the reserved cache line is set to the shared state in response to a writeback transaction from the remote processing node to the home processing node. The examiner also cites column 16, lines 16-24 at Baumgartner. This section is directed to claim 3 and portions of claim 2 which claim the concept of a writeback transaction.

These sections of Baumgartner have nothing to do with relinquishing exclusive control of a memory block in response to commencing but not completing write

operations to the memory block as recited in claim 21. Furthermore, these sections of Baumgartner have nothing whatsoever to do with a processing node relinquishing exclusive control of the memory block and then transmitting a message that indicates that the Owner processor should still be deemed a **sharer of memory**. In Baumgartner, no such signal is transmitted. Instead, the writeback transaction in Baumgartner is used to set the coherence state of the cache line to non-exclusive or shared. The signal is not indicating that the remote processor remains a sharer of memory. Again, the signal prompts the coherence state to be set. Furthermore, Baumgartner never teaches that the action of transmitting the writeback is "in response to commencing but not completing write opeations" as recited in claim 21.

The differences between the claims and the teachings in the art are great since the cited references fail to teach or suggest all of the claim elements. As such, the pending claims are not a predictable variation of the art to one of ordinary skill in the art.

For at least these reasons and the reasons provided above with respect to independent claim 1, 13 and 24, claims 2, 3, 14, 15, 21, 22, 25 and 26 are allowable over the cited art of record.

Dependent Claim 15

Dependent claim 15 recites that the Load Lock instruction forms part of a Load Lock/Store Conditional instruction pair. The art of record does not teach or suggest this claim element.

The examiner previously admitted that Hagersten does not teach this element (see Final OA mailed 07/26/06 at p. 5). Appellants agree with this admission. In the Final OA mailed 04/04/2008, the examiner has not cited another reference for allegedly teaching this claim element. In view of the admissions in the Final OA mailed 07/26/2008, the examiner has not established a prima facie case of obviousness for claim 15.

Claim Rejections: 35 USC § 103(a)

Claims 4, 5, 7, 8, 18, 19, and 28-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 5,887,138 to Hagersten et al. in view of U.S. Pat. No. 6,275,907 to Baumgartner et al. and further in view of U.S. Pat. No. 6,141,734 B1 to

Razdan et al. and further in view of U.S. Pat. No. 6,185,658 B1 to Arimilli et al. These rejections are traversed.

Claims 4, 5, 7, 8, 18, 19, and 28-30 are allowable for at least the reasons provided above with respective independent claims 1, 13, and 24.

Claim Rejections: 35 USC § 103(a)

Claims 9 and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 5,887,138 to Hagersten et al. in view of U.S. Pat. No. 6,275,907 to Baumgartner et al. and further in view of U.S. Pat. No. 6,141,734 B1 to Razdan et al. and further in view of U.S. Pat. No. 6,185,658 B1 to Arimilli et al and further in view of U.S. Pat. No. 6,266, 744 to Hughes et al. These rejections are traversed.

Claims 9 and 31 are allowable for at least the reasons provided above with respective independent claims 1 and 24.

CONCLUSION

In view of the above, Appellants respectfully request the Board of Appeals to reverse the Examiner's rejection of all pending claims.

Any inquiry regarding this Amendment and Response should be directed to Philip S. Lyren at Telephone No. 832-236-5529. In addition, all correspondence should continue to be directed to the following address:

Hewlett-Packard Company
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

Respectfully submitted,

/Philip S. Lyren #40,709/

Philip S. Lyren
Reg. No. 40,709
Ph: 832-236-5529

VIII. Claims Appendix

1. A distributed multiprocessing computer system, which includes a plurality of processors each coupled to an associated memory module, wherein each of the associated memory modules stores data that is shared between said processors, said system comprising:

a Home processor that includes a memory block and a directory for said memory block in the associated memory module;

an Owner processor that includes a cache memory, and wherein said Owner processor obtains an exclusive copy of said memory block, and stores said exclusive copy of said memory block in said cache memory; and

wherein said Owner processor begins write operations on said memory block and then displaces the exclusive copy of said memory block from said cache memory prior to completing the write operations on said memory block, and in response to displacing said memory block prior to completing the write operations said Owner processor returns said displaced copy of said memory block to said Home processor with a signal indicating that said Owner processor remains a sharer of said memory block.

2. The distributed multiprocessing computer system of claim 1, wherein said Owner processor obtains an exclusive copy of said memory block by issuing a Load Lock instruction, and wherein the directory associated with the Home processor indicates that said Owner processor has obtained exclusive control of said memory block.

3. The distributed multiprocessing computer system of claim 2, wherein said Owner processor executes multiple threads concurrently, and displaces data associated with a non-executing thread from its associated cache memory.

4. The distributed multiprocessing computer system of claim 3, wherein said Owner processor includes a register in which an address is stored representing the memory block obtained in response to the Load Lock instruction, and wherein said Owner processor compares the address of any displaced data with the address stored in said register.

5. The distributed multiprocessing computer system of claim 4, wherein the Owner processor asserts a Victim To Shared message if the address of any displaced data matches the address stored in said register.

6. The distributed multiprocessing computer system of claim 5, wherein the Owner processor asserts a Victim message if the address of any displaced data does not match the address stored in said register.

7. The distributed multiprocessing computer system of claim 1, wherein a directory associated with the Home processor indicates that said Owner processor has become a sharer of said memory block in response to receiving said signal from said Owner processor.

8. The distributed multiprocessing computer system of claim 7, wherein said Owner processor subsequently re-obtains an exclusive copy of said memory block to complete execution of the non-executing thread.

9. The distributed multiprocessing computer system of claim 8, wherein the Owner processor asserts a Read-with-Modify Intent Store Conditional instruction to the Home directory to again request an exclusive copy of said memory block.

10. The distributed multiprocessing computer system of claim 9, wherein, in response to the Read-with-Modify Intent Store Conditional instruction, the Home directory determines if the Owner processor is a sharer of the memory block, and if so, the Home directory sends an exclusive copy of the memory block to the Owner processor.

11. The distributed multiprocessing computer system of claim 10, wherein the Home directory invalidates all other sharers when it sends an exclusive copy of the memory block to the Owner processor.

12. The distributed multiprocessing computer system of claim 9, wherein the Home directory determines if the Owner processor is a sharer of the memory block, and if not, the Home directory sends a Store Conditional Failure message to the Owner processor.

13. A method of maintaining memory coherence in a distributed shared memory computer system including a plurality of processors, comprising:

requesting a copy of a memory block from a Home processor to perform a write operation on the copy of the memory block;

storing said copy of said memory block exclusively in a cache memory associated with an Owner processor;

updating a coherence directory for said memory block in said Home processor to indicate the write operation on the copy of said memory block in said cache memory associated with the Owner processor;

displacing said copy of said memory block from said cache memory prior to completion of said write operation on said memory block;

in response to commencing but not completing said write operation, transmitting a message from said Owner processor to said Home processor relinquishing exclusive control of said memory block, said message indicating that said Owner processor should still be deemed a sharer of said memory block.

14. The method of claim 13, wherein the copy of the memory block is requested using a Load Lock instruction from the Owner processor to the Home processor.

15. The method of claim 13, wherein the Load Lock instruction forms part of a Load Lock/Store Conditional instruction pair.

16. The method of claim 13, wherein the updating the coherence directory includes modifying a register to indicate that the Owner processor has an exclusive copy of the memory block.

17. (Canceled)

18. The method of claim 13, wherein the transmitting a message includes assertion of a Victim To Shared message if an address of the displaced memory block matches an address of any memory block for which an exclusive copy resides in the Owner processor, and wherein the coherency directory associated with the Home processor indicates that said Owner processor has become a sharer of said memory block in response to said Victim To Shared message.

19. The method of claim 18, further comprising the updating the coherence directory to indicate that said Owner processor has become a sharer of said memory block in response to said Victim To Shared message.

20. (Canceled)

21. A method of maintaining memory coherence in a distributed shared memory computer system including a plurality of processors, comprising:

requesting a copy of a memory block from a Home processor to perform write operations on the copy of the memory block;

storing said copy of said memory block exclusively in a cache memory associated with an Owner processor;

updating a coherence directory for said memory block in said Home processor to indicate the write operation on the copy of said memory block in said cache memory associated with the Owner processor;

displacing said copy of said memory block from said cache memory prior to completion of the write operations on said memory block;

in response to commencing but not completing said write operations, transmitting a message from said Owner processor to said Home processor relinquishing exclusive control of said memory block, said message indicating that said Owner processor should still be deemed a sharer of said memory block;

asserting a request to again obtain an exclusive copy of said memory block, wherein, in response to the request to again obtain an exclusive copy of the memory block, the Home processor determines if the Owner processor is a sharer of the memory block, and if so, the Home processor sends an exclusive copy of the memory block to the Owner processor.

22. The method of claim 21, wherein the Home processor invalidates all other sharers when it sends an exclusive copy of the memory block to the Owner processor.

23. The method of claim 21, wherein, in response to the request to again obtain an exclusive copy of the memory block, the Home processor directory determines if the

Owner processor is a sharer of the memory block, and if not, the Home directory sends a Store Conditional Failure message to the Owner processor.

24. A distributed multiprocessing computer system, comprising:

a first processor that includes a memory block and a directory associated with said memory block that tracks a status of said memory block;

a second processor that includes a cache memory, and wherein said second processor requests an exclusive copy of said memory block and stores said memory block in said cache memory; and

wherein said second processor begins processing of said memory block and then displaces the exclusive copy of said memory block prior to completing the processing of said memory block, and in response to displacing said memory block but not completing the processing said second processor transmits a signal to said first processor indicating that said second processor relinquishes exclusive control of said memory block but should remain a sharer of said memory block.

25. The distributed multiprocessing computer system of claim 24, wherein said second processor obtains an exclusive copy of said memory block by issuing a Load Lock instruction, and wherein the directory associated with the first processor indicates that said second processor has obtained exclusive control of said memory block.

26. The distributed multiprocessing computer system of claim 24, wherein said second processor executes multiple threads concurrently, and displaces data associated with a non-executing thread from its associated cache memory.

27. (Canceled)

28. The distributed multiprocessing computer system of claim 24, wherein the second processor asserts a Victim To Shared message if the address of any displaced data matches the address stored in said register.

29. The distributed multiprocessing computer system of claim 28, wherein the directory associated with the first processor indicates that said second processor has become a sharer of said memory block in response to said Victim To Shared message.

30. The distributed multiprocessing computer system of claim 24, wherein said second processor subsequently re-obtains an exclusive copy of said memory block from said first processor to complete processing of said memory block.

31. The distributed multiprocessing computer system of claim 30, wherein the second processor asserts a request to read, modify, and conditionally store said memory block to said first processor.

32. The distributed multiprocessing computer system of claim 31, wherein, in response to the request to read, modify, and conditionally store said memory block, the first processor determines if the second processor is a sharer of the memory block, and if so, the first processor sends an exclusive copy of the memory block to the second processor.

33. The distributed multiprocessing computer system of claim 32, wherein the first processor invalidates all other copies of said memory block when it sends an exclusive copy of the memory block to the second processor.

34. The distributed multiprocessing computer system of claim 31, wherein, in response to the request to read, modify, and conditionally store said memory block, the first processor determines if the second processor is a sharer of the memory block, and if not, the first processor sends a failure message to the second processor.

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None.